

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION**

BLUE SPIKE, LLC,

Plaintiff,

v.

BLU PRODUCTS, INC., *et al.*,

Defendants.

Case No. 6:16-cv-271-RWS-JDL

LEAD CASE

BLUE SPIKE, LLC,

Plaintiff,

v.

TOSHIBA AMERICA INFORMATION
SYSTEMS, INC. & TOSHIBA
CORPORATION

Defendants.

Case No. 6:16-cv-430-RWS-JDL

CONSOLIDATED CASE

DEFENDANTS' RESPONSIVE CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. APPLICABLE LAW	3
III. LEVEL OF ORDINARY SKILL IN THE ART	3
IV. DISPUTED TERMS	4
a. “a software application”; “application”	4
b. “a program”	7
c. “copy protecting a software application”	10
d. “during execution of the software application”; “when executing said software application”	12
e. “memory scheduler”; “memory scheduler code resource”	15
f. “call”	19
g. “intermittently relocating”; “intermittently”; “relocating”	20
h. “shuffle” “randomize” “relocate”	23
V. CONCLUSION	26

TABLE OF AUTHORITIES

	<u>Page(s)</u>
Cases	
<i>Applied Med. Res. Corp. v. U.S. Surgical Corp.</i> , 448 F.3d 1324 (Fed. Cir. 2006).....	13, 24
<i>Bicon, Inc. v. Straumann Co.</i> , 441 F.3d 945 (Fed.Cir.2006).....	22
<i>Colby v. J.C. Penney Co.</i> , 811 F.2d 1119 (7th Cir. 1987)	1
<i>Cultor Corp. v. A.E. Staley Mfg. Co.</i> , 224 F.3d 1328 (Fed. Cir. 2000).....	17
<i>Daiichi Sankyo Co. Ltd. v. Apotex, Inc.</i> , 501 F.3d 1254 (Fed. Cir. 2007).....	3
<i>Enzo Biochem Inc. v. Applera Corp.</i> , 780 F.3d 1149 (Fed. Cir. 2015).....	22
<i>Fargo Elecs., Inc. v. Iris Ltd., Inc.</i> , No. 04-1017	24
<i>Iridescent Networks, Inc. v. AT&T Mobility, LLC</i> , Civ. No. 6:16-CV-01003-RWS (E.D. Tex. April 3, 2017).....	2
<i>Iris Connex, LLC v. Dell, Inc.</i> , Civ. No. 2:15-CV-1915-JRG, D.I. 149 (E.D. Tex. January 25, 2017)	2
<i>Jonsson v. Stanley Works</i> , 903 F.2d 812, 14 USPQ2d 1863 (Fed. Cir. 1990)	14
<i>KX Indus., L.P. v. PUR Water Purification Prod., Inc.</i> , 108 F. Supp. 2d 380 (D. Del. 2000).....	1
<i>Markman v. Westview Instr., Inc.</i> , 517 U.S. 370, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996).....	1
<i>Markman v. Westview Instruments, Inc.</i> , 52 F.3d 967 (Fed. Cir. 1995) (<i>en banc</i>), <i>aff’d</i> , 517 U.S. 370 (1996).....	3
<i>MyMedicalRecords, Inc. v. Walgreen Co.</i> , No. 2:13-CV-00631-ODW, 2014 WL 4367949 (C.D. Cal. Sept. 3, 2014)	20

<i>MySpace, Inc. v. GraphOn Corp.</i> , 672 F.3d 1250 (Fed. Cir. 2012).....	3
<i>Noah Sys., Inc. v. Intuit Inc.</i> , 675 F.3d 1302 (Fed. Cir. 2012).....	18
<i>O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.</i> , 521 F.3d 1351 (Fed. Cir. 2008).....	3, 15
<i>Oasis Research, LLC v. AT & T Corp.</i> , No. 4:10-CV-00435, 2012 WL 602199 (E.D. Tex. Feb. 23, 2012).....	9
<i>Omega Eng’g, Inc. v. Raytek Corp.</i> , 334 F.3d 1314 (Fed. Cir. 2003).....	14
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	3, 9
<i>Seachange Int’l, Inc. v. C-Cor Inc.</i> , 413 F.3d 1361 (Fed. Cir. 2005).....	5
<i>Tex. Dig. Sys., Inc. v. Telegenix, Inc.</i> , 308 F.3d 1193 (Fed. Cir. 2002).....	5
<i>Triton Tech of Texas, LLC v. Nintendo of Am., Inc.</i> , 753 F.3d 1375 (Fed. Cir. 2014).....	19
<i>Vitronics Corp. v. Conceptronic, Inc.</i> , 90 F. 3d 1576 (Fed. Cir. 1996).....	3, 9
<i>Williamson v. Citrix Online, LLC</i> , 792 F.3d 1339 (Fed. Cir. 2015).....	16, 17, 18, 19
<i>Zenon Envtl., Inc. v. U.S. Filter Corp.</i> , 506 F.3d 1370 (Fed. Cir. 2007).....	11

Statutes

35 U.S.C. § 102(a)	2
--------------------------	---

Other Authorities

Local Rule CV-5	28
Local Rule CV-5(a)(3)(A)	28
U.S Patent 7,664,263	2
U.S. Patent 9,104,842	8, 14

U.S. Patent No. 5,613,004.....11

U.S. Patent No. 5,745,569.....1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 22, 23, 25, 26

U.S. Patent No. 8,930,719.....1, 2, 5, 8, 9, 14, 17, 18, 19, 20

I. INTRODUCTION

Plaintiff Blue Spike, LLC (“Blue Spike”) asserts claims 16-20 of U.S. Patent No. 5,745,569 (“‘569 Patent”) and claims 10-17, 22, and 23 of U.S. Patent No. 8,930,719 (“‘719 Patent”)(collectively “Asserted Patents” or “Asserted Claims”) against Toshiba America Information Systems, Inc. and Toshiba Corporation (collectively “Toshiba”). The parties dispute constructions of ten claim terms.

In *Blue Spike, LLC v. Huawei Technologies CO., LTD*, et al., 6:13-cv-00679-RWS (D.I. 194 – Ordered May 16, 2016) (Ex. 4 “Huawei Decision”) this Court rejected most of the claim construction positions that Blue Spike advances again here, a fact that Blue Spike never addresses in its claim construction brief. One of those constructions, for “software application/application,” disposes of all asserted claims in the ‘569 patent and all but two of the asserted claims in the ‘719 patent. A second term — “program” — that was not previously construed, but is indistinguishable from the construed term “software application,” disposes of these remaining two claims. These constructions are dispositive because Blue Spike’s infringement allegations against Toshiba are directed to the Android operating system, which is not a software application or program within the meaning of these patents. That means a crucial element of Blue Spike’s infringement proof is missing for every claim.

Toshiba respectfully submits that this Court must give considerable weight to its previous constructions of these same terms. *Colby v. J.C. Penney Co.*, 811 F.2d 1119, 1123 (7th Cir. 1987)(“The most complex relationship is between a court and its own previous decisions. A court must give considerable weight to those decisions”); *KX Indus., L.P. v. PUR Water Purification Prod., Inc.*, 108 F. Supp. 2d 380, 387 (D. Del. 2000), *aff’d sub nom. KX Indus. v. PUR Water Purification Prod., Inc.*, 18 F. App’x 871 (Fed. Cir. 2001); *See Markman v. Westview*

Instr., Inc., 517 U.S. 370, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996) (noting that generally that courts should promote uniformity in the claim construction of a given patent). Blue Spike's complete failure to address why this Court should depart from its prior precedent should foreclose further dispute about the meanings of those terms and any arguments that Blue Spike makes in reply on this subject should be ignored because Blue Spike knew from the parties' Joint Claim Construction Statement that Toshiba cited the prior rulings in support of its positions here and yet Blue Spike chose not to address this issue. D.I. 57.

Assuming this Court does follow its precedent, Toshiba respectfully requests the opportunity to submit a short, focused motion for summary judgment of both non-infringement and invalidity.¹ That motion will show that (1) all of the asserted claims of the '569 Patent should be dismissed because Blue Spike does not accuse any software application, as is required by all asserted claims, and (2) that all of the asserted claims of the '719 Patent should be dismissed both because of this Court's constructions and because Blue Spike copied the prior art '569 Patent disclosure into the '719 Patent and disclaimed² any priority to the '569 Patent, thus rendering the asserted claims of the '719 Patent anticipated by the '569 Patent.³

¹ An early and expedited motion for summary judgment may be appropriate after claim construction. *See, e.g., Iridescent Networks, Inc. v. AT&T Mobility, LLC*, Civ. No. 6:16-CV-01003-RWS (E.D. Tex. April 3, 2017); *Iris Connex, LLC v. Dell, Inc.*, Civ. No. 2:15-CV-1915-JRG, D.I. 149 (E.D. Tex. January 25, 2017).

² *See e.g., '719 Patent prosecution history, preliminary amendment Sep. 8, 2012 at 4* ("the specification is amended to remove the priority claim to [] U.S. Pat. No. 5,745,569"); U.S Patent 7,664,263 prosecution history (a predecessor patent to the '719 patent), request for certificate of correction, dated Feb. 6, 2011 (requesting certificate of correction to remove "erroneous" priority claim to the '569 patent).

³ The '569 patent is prior art to the '719 patent under at least 35 U.S.C. § 102(a) and/or 102(e). Toshiba served Blue Spike a first set of interrogatories (1-12) on February 27, 2017, which included a request for Blue Spike to "identify the earliest priority date for each claim." (Ex. 1, Toshiba's First Set of Interrogatories at 7). Blue Spike failed to provide any response or objection to any interrogatory within the

II. APPLICABLE LAW

“When the parties present a fundamental dispute regarding the scope of a claim term, it is the court’s duty to resolve it.” *O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1360 (Fed. Cir. 2008). When construing claims of a patent, courts consider “what was invented and what exactly was claimed.” *MySpace, Inc. v. GraphOn Corp.*, 672 F.3d 1250, 1256 (Fed. Cir. 2012).

The starting point for a claim construction inquiry is the claims themselves. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (*en banc*). The “claims must be read in view of the specification, of which they are a part.” *Id.* at 1315 (*quoting Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 978 (Fed. Cir. 1995) (*en banc*), *aff’d*, 517 U.S. 370 (1996)). The specification “is the single best guide to the meaning of a disputed term.” *Id.* (*quoting Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)).

III. LEVEL OF ORDINARY SKILL IN THE ART

In determining the level of ordinary skill as it pertains to an invention, the court considers many factors such as “(1) the educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of active workers in the field.” *Daiichi Sankyo Co. Ltd. v. Apotex, Inc.*, 501 F.3d 1254, 1256 (Fed. Cir. 2007).

Considering these factors, Toshiba agrees with Blue Spike that one of ordinary skill in the art would have a Master’s degree in computer science or computer engineering, or equivalent experience. Contrary to Blue Spike’s additional requirements, one of ordinary skill in the art

deadline set by the Federal Rules. The Parties met and conferred on April 4, 2017 and Blue Spike responded that they would “look into it.” Toshiba has yet to receive any response.

would not additionally have had two years of experience in the field of digital fingerprinting and cryptography. (*See* D.I. 76 at 4). Blue Spike has not offered any explanation as to why experience in digital fingerprinting and cryptography would be necessary.

IV. DISPUTED TERMS

All of Toshiba's proposed constructions of the '569 Patent terms are the same as those already construed by Judge Schroeder. *See Blue Spike, LLC v. Huawei Technologies CO., LTD*, et al., 6:13-cv-00679-RWS (D.I. 194 – Ordered May 16, 2016).

a. “a software application”; “application”

Defendants' Construction	Blue Spike's Construction
Proposed construction: “a software program run by an operating system”	Proposed construction: “a computer program” Alternate construction: Plain and ordinary

Toshiba's proposed construction is Judge Schroeder's construction from the *Huawei* case. This construction appeared in the parties' Joint Claim Construction Chart. Blue Spike does not address the potential *stare decisis* effect. This is significant because the construction disposes of all but two asserted claims, as explained *supra*. In the absence of any argument by Blue Spike that *stare decisis* does not apply, there is no need for this Court to consider this term further.

Should this Court decide to analyze the term further, Toshiba respectfully submits that the claim language demonstrates the proper construction of “software application” by distinguishing “computer software” (that may include an operating system and software applications) from “software application.” For example, claim 1 of the '569 Patent discloses a method “for copy protection of computer software” ('569 Patent at 8:30-31) while Claim 16 of

the ‘569 Patent sets out a method for protecting a “software application” comprising a plurality of executable code resources. *See* ‘569 Patent at 10:8-12. Additionally, Claim 1 of the ‘719 Patent recites “[a] computing device for running application software” and further requires “an operating system” as a separate claim element from “said application software.” ‘719 patent at 16:2-16. The inventors’ use of “operating system” and “application,” separately, not interchangeably, evidences that they intended the two terms to be distinct. *See Tex. Dig. Sys., Inc. v. Telegenix, Inc.*, 308 F.3d 1193, 1209 (Fed. Cir. 2002).

In yet another example, claim 3 of the ‘719 patent refers to “said **operating system** is designed to enable **application software** to modify a value stored by said program counter.” ‘719 Patent at 16:37-40 (emphases added). This distinction appears in the specification as well where the inventors distinguish between times when “system code” is running as distinct from “application code.” (‘569 Patent at 7:30-34)⁴ (“[code resources] typically remain in a fixed position [in memory], unless the computer **operating system** finds it necessary to rearrange certain portions of memory during “system time,” when the **operating system code, not application code**, is running.”) (emphases added).

Thus, in light of the specification and claims of the Asserted Patents, the term “software application” cannot be construed as Blue Spike contends – as including an operating system. *See Seachange Int’l, Inc. v. C-Cor Inc.*, 413 F.3d 1361, 1368 (Fed. Cir. 2005) (“The doctrine of claim differentiation stems from ‘the common sense notion that different words or phrases used in separate claims are presumed to indicate that the claims have different meanings and scope.’”)

Judge Schroeder’s and Toshiba’s construction is also consistent with extrinsic sources that show a person of skill in the art would have understood the term “software application” not

⁴ Defendants note that the specification of the ‘569 patent appears word-for-word in the ‘719 patent. The specification portions relevant to the asserted claims are the same. Therefore, citations to the specifications will be made to the ‘569 patent only unless where a specific ‘719 claim is discussed.

to include operating systems. For example, the Dictionary of Computer and Internet Terms, published in 1998 defines “application program” to be “a computer program that performs useful work not related to the computer itself.” Examples include word processors, spreadsheets, accounting systems, and engineering programs. (Ex. 2, Douglas A. Downing, Michael A. Covington, Melody Mauldin Covington, Dictionary of Computer and Internet Terms, 22 (Barron’s Educational Series, Inc., 6th ed. 1998) (emphasis added).) Numerous other dictionaries confirm that a person of skill in the art plainly would have understood that software applications are distinct from operating systems. (*See, e.g.*, Ex. 3, New Oxford American Dictionary Third Edition 1229 (Oxford University Press 2010) (“**operating system** is the software that supports a computer’s basic functions, such as scheduling tasks, **executing applications**, and controlling peripherals.”) (emphasis added)).

Both this intrinsic and extrinsic evidence caused Judge Schroeder to conclude that operating systems are distinct from software applications and are used to facilitate running software applications. Ex. 4 Huawei Decision at 11.

Blue Spike’s proposed constructions should be rejected for two additional reasons. First, Blue Spike claims that the inventors acted as their own lexicographer but cites to no portion of the specification where the inventors defined this limitation. Blue Spike’s only other support for its proposed construction is a sentence in the specification that discusses the background of the invention and how a *user* views an application. Blue Spike’s selection of part of this sentence seeks to mislead this Court claiming the specification discloses an application as a computer program. This is not true. The sentence Blue Spike refers to – that states “an executable computer program” is referred to as an application “from the point of view of a user” – continues on to explain that it is viewed as executable code from the point of view of an engineer. ‘569

Patent at 3:44-47; Id. at 4:60-65. Executable code, i.e., referring to a software program run by an operating system – is the construction proposed by Toshiba. D.I. 76 at 12. The intrinsic record thus supports that applications and programs are run by an operating system in the context of the Asserted Patents. There is no support for Blue Spike’s proposed construction in the specification.

Further, Blue Spike uses a dictionary definition from 1972—24 years before the ’569 patent was filed—to support its position that a software application can be any computer software. In between the time that Blue Spike’s dictionary was published and the filing of the ’569 patent, there were a great number of developments in the computing world including various operating systems released publicly such as Unix, Apple DOS, Microsoft Dos, Windows, Linux, that render Blue Spike’s proposed construction obsolete.

b. “a program”

Defendants’ Construction	Blue Spike’s Construction
<u>Proposed construction:</u> “a set of instructions run by an operating system” Alternate construction: “A software program run by an operating system” ⁵	<u>Proposed construction:</u> Plain and ordinary

Judge Schroeder did not construe this term. The term appears in claims 22 and 23 of the ’719 patent and, like “software application” it does not include an operating system.

Turning first to claim 22, the claim language notably explains that a [computer] system is configured to “load an executable program” into memory. As this Court already decided in

⁵ This alternate construction was not included in the Joint Claim Construction Statement. Toshiba offers it now for the first time in this brief based on Blue Spike’s opening arguments that contradict the specification.

Huawei, loading precedes execution. Ex. 4 Huawei Opinion at 7. Thus, the executable program must first be loaded before it can be executed. And “loading” is the act of transferring the program from storage into memory. *Id.* Without question, the operating system performs the loading; it is the only software resource that is available to perform that task. It is therefore part of claim 22’s system and it is not a “program.” This distinction applies equally to claim 23’s preamble which recites a “system for executing a program.” To execute a program, the operating system must first load the program into memory.

Like the claim language, the specification consistently reflects this distinction between program and system or operating system. First, the specification explains that a preferred embodiment would be “implemented in an embedded system with a minimal operating system.” ‘719 Patent at 14:35-38. Here, “embodiment,” commonly understood in patents to mean the invention, is described separately and therefore is plainly distinct from an operating system.

This distinction appears again shortly afterwards when the specification explains that “[o]nce the code resources of a **program** are loaded into memory, they typically remain in a fixed position, unless the computer **operating system** finds it necessary to rearrange certain portions of memory during ‘system time,’ when the **operating system** code, **not application code** is running.” ‘569 Patent at 7:30-34, emphases added⁶. And again, only a few lines farther down, the specification states that the **operating system** “rearrange[s] memory transparently, underneath a running **program**.” ‘569 Patent at 7:38-39 (emphases added). In every one of these passages, the patentees clearly distinguish “system” and “operating system” from “program.” Finally, the specification states that “the present invention concerns itself with any application software that may be used in general computing devices” ‘569 Patent at 7:59-

⁶ Indeed, the Applicant relied on this passage in responding to an Office Action in the direct precedent application to the ‘719 patent. *See* Ex. 5 U.S. Patent 9,104,842 Prosecution history, Response dated 2011/02/28, pgs. 3-4.

61. In claims 22 and 23 of the '719 patent, that "present invention" is the program that randomizes location of code. In other words, that program is "application software" that may be used in general computing devices that already have operating systems. Indeed, the specification equates "program" and "application" and explains that to an engineer, a "**computer program** is variously referred to as **an application**, from the point of view of a user," meaning that the referenced computer program is an application, and not an operating system. '569 Patent at 3:44-45(emphases added).

If "a program" were to encompass an operating system as Blue Spike suggests, then none of these passages in the specification would be accurate. A construction that conflicts with both the claim language and with numerous, consistent passages from the specification is not correct. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) ("claims must be read in view of the specification, of which they are a part. As we stated in *Vitronics*, the specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.'")(internal citations omitted); *Oasis Research, LLC v. AT & T Corp.*, No. 4:10-CV-00435, 2012 WL 602199, at *24 (E.D. Tex. Feb. 23, 2012) (rejecting construction that conflicts with the specification). In these Asserted Patents, at least, both "program" and "software application" are synonymous with each other and are distinct from operating systems.

The very dictionary Blue Spike relies on, Microsoft Computer Dictionary 5th Edition 2002, confirms this analysis: a program is defined as "A sequence of instructions that can be executed by a computer. The term can refer to the original source code or to the executable (machine language) version." See Ex. 6 Microsoft Computer Dictionary 5th Edition 2002 at 424. Plainly, the operating system, which controls the inner workings of the computer and

applications, causes the execution of any such program. This language is synonymous with the language used in the specification of the '569 Patent. Accordingly, Toshiba's proposed construction should be adopted. In the alternative, "program" should be construed the same as "application" because the specification equates them.

c. "copy protecting a software application"

Defendants' Construction	Blue Spike's Construction
<u>Proposed construction:</u> "protecting a software application from unauthorized copying or the use of unauthorized copies"	<u>Proposed construction:</u> Plain and ordinary <u>Alternate construction:</u> "protecting against the unauthorized copying, unauthorized use, or unintended use of a software application"

Judge Schroeder not only adopted Toshiba's proposed construction of this term in *Blue Spike, LLC v. Huawei Technologies CO., LTD*, et al., 6:13-cv-00679-RWS (D.I. 194 – Ordered May 16, 2016), but he did so after Blue Spike agreed to that construction. Ex. 4 Huawei Decision at 5.⁷ Now Blue Spike recants its prior agreement, alluding to an alleged misrepresentation of the construction by Huawei in that prior case. A third party's alleged misunderstanding of a construction is hardly a reason to vary a claim construction and Blue Spike never bothers to explain this alleged misunderstanding.

The '569 patent specification uses the term in accordance with its plain and ordinary meaning, i.e. to protect a software application from unauthorized copying. The specification of the '569 Patent states "distribution and exchange of content would be made more secure from unauthorized copying," and that the disclosed invention prevents "unauthorized copies" and prevents hackers from "forc[ing] the application program to run as an unauthorized copy." '569

⁷ Blue Spike does not dispute that the preamble is a limitation either in the Joint Claim Construction Chart or in its brief. Accordingly, there is no dispute between the parties on that issue and Toshiba does not address it here.

Patent at 4:49- 53, 5:51-57, 7:13-17. Indeed, the summary of the invention conveys that the purpose of the invention is to protect the copyright owner from unauthorized copying. '569 patent at 3:30-41.

Further, the '569 patent incorporates by reference U.S. Patent No. 5,613,004, entitled "Steganographic Method and Device" and issued to the same inventors (the "'004 patent") (See '569 patent at 2:34-44) and thus includes the disclosure of the '004 patent. *Zenon Envtl., Inc. v. U.S. Filter Corp.*, 506 F.3d 1370, 1378 (Fed. Cir. 2007). The '004 patent specification also identifies "copy protection" as being directed to the prevention of making unauthorized copies. (Ex. 7 ('004 patent) at 2:57-60.) Moreover, the '569 prosecution history shows the applicants argued, to secure allowance of its claims, that "a significant benefit of the claimed invention" is that it protects software from unauthorized copying. Ex. 8 ('569 Patent prosecution history, Response to Office Action dated July 23, 1997 at 2-3).

Extrinsic evidence also supports Defendants' construction. The dictionary definition of "copy protecting" is "[r]esistance to unauthorized copying of software." (See Ex. 9 (The Computer Glossary, AMACOM, at 80 (1998))). Even the dictionary relied upon by Blue Spike confirms that Defendants' construction is correct. (See Ex. 10, Microsoft Computer Dictionary, Fifth Edition 2002 ("copy protection n. A software lock placed on a computer program by its developer to prevent the product from being copied and distributed without approval or authorization.")).

Blue Spike claims this term should be construed by its plain and ordinary meaning because it is "readily understood" and then says that "the potential for confusion is real" with regard to interpreting this limitation. D.I. 76 at 5-6. This is nonsensical. Blue Spike points to the *Huawei* construction as creating confusion, without any citations to evidence, and concludes

that the “prior construction has proven inadequate” asserting that Huawei made incorrect assumptions. Blue Spike never explains its position on this point any further. Additionally, Blue Spike cites to other potential definitions for copy protection, but cites to no evidence to support these definitions.

Further, Blue Spike claims that its alternative proposal “does not imply that copy protection is only relevant if a copy has been made” but it contends Toshiba’s construction is so limited. Toshiba’s construction is not so limited: it expressly states that the protection is also against “use of unauthorized copies.” In fact, this construction finds support in the embodiment that Blue Spike cites for its alternative construction (‘569 Patent at 2:21-26), where the ‘569 patent discusses the problem of a malicious user disabling “countermeasures against unlicensed **copying**” “by means of a ‘patch,’” a problem asserted to be solved by the present application. ‘569 Patent at 7:39-44 (emphasis added); *See also Id.* at 3:29-30 (“[w]ithout the correct license information, the copy cannot function”—i.e., to disable unauthorized copies). Blue Spike’s cluttered alternate construction goes beyond the intrinsic evidence when the specification clearly states the intended function of the “copy protection” feature. Toshiba’s construction honors the specification, and should be adopted by this Court.

d. “during execution of the software application”; “when executing said software application”

Defendants’ Construction	Blue Spike’s Construction
<u>Proposed construction:</u> “while the software application is running”	<u>Proposed construction:</u> Plain and ordinary <u>Alternate construction:</u> “the time between when an application begins and ends”

Toshiba's proposed construction is Judge Schroeder's previous construction in the *Huawei* case. As an initial matter, Claim 16 of the '569 patent distinguishes between loading and execution of the software application. The preamble of the claim refers to resources that have already been loaded (i.e. "plurality of code resources **loaded** in a memory"), and then refers to a "software application" and "executable code resources" – indicating that the code resources have the potential to be executed in the future. '569 patent at 10:10-12 (emphasis added). The temporal nature of the language requires that the executable code resources already be loaded into memory before execution takes place. Indeed, loading and execution are different claim terms, and different claim terms are presumed to have different meanings. *Applied Med. Res. Corp. v. U.S. Surgical Corp.*, 448 F.3d 1324, 1333 n. 3 (Fed. Cir. 2006).

Claim 16 then refers to "intermittently relocating each of the plurality of executable code resources to a different address within the memory of the computer **during execution** of the software application." '569 patent at 10:16-19, (emphasis added). Thus, "during **execution** of the software application" and "when **executing** said software application" does not refer to load time; the patent claims delineate loading ("loaded in memory") as a separate operation from execution.

Even according to the Applicant, the Asserted Patents were allegedly inventive over the prior art because the invention rearranged code resources after loading and *while* the application was running instead of during system time. '569 Patent at 8:1-9 (describing "intentional[] shuffl[ing]" of executable code resources "[d]uring execution time" at "periodic[]" or "random or pseudo-random intervals."). During the prosecution of the direct predecessor⁸ to the '719

⁸ Statements regarding terms in related applications are imported into current constructions. See *Omega Eng'g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1333 (Fed. Cir. 2003) ("prosecution disclaimer may arise from disavowals made during the prosecution of ancestor patent applications"); *Jonsson v. Stanley Works*, 903 F.2d 812, 818, 14 USPQ2d 1863, 1869 (Fed. Cir. 1990) (holding that when two patents issued from

patent, the applicant distinguished two prior art references arguing: “Neither [prior art reference] disclosed embedding data into software **while running the software** . . .” and therefore neither prior art reference disclosed performing this step “. . . **during execution of the software.**” (Ex. 11 U.S. Patent 9,104,842, Prosecution History at 83 of 114.) (emphases added). Applicant himself equated “execution” with “running.”

Toshiba’s proposed construction is also in accord with the extrinsic evidence. Multiple technical dictionaries explain that a program’s execution refers to running the instructions within the program itself. *See e.g.*, Ex. 12 George McDaniel, IBM Dictionary of Computing 250 (McGraw-Hill, Inc. 1994) (“Execute: To perform the actions specified by a program or a portion of a program.”). Technical dictionaries also distinguish between execution and:

loading. Compare Ex. 13 Dictionary of Computer Words, Am. Heritage Dictionaries Revised Ed., at 94 (1995) (“**Execute** : To run a program, carry out a command, or perform a function.) with *Id.* at 160 (**load**: to transfer a program from a storage device into a computer’s memory.).

Blue Spike’s contention that plain and ordinary meaning applies here and Blue Spike’s alternate construction are equally inappropriate. Blue Spike’s alternative proposed construction is contrary to the intrinsic record that distinguishes loading from execution. Further, Blue Spike provides an example to support its alternative construction without any source – that double clicking on an application means that “an application is executed.” D.I. 76 at 10. Just because a “user double clicks on an application,” (as recited in Blue Spike’s example) which may cause the loading and the successive executing of the application, does not mean the two are the same. To

continuation-in-part applications derived from one original application, the prosecution history of a claim limitation in the first patent to issue was properly applied to the same claim limitation in the second patent to issue).

the contrary, loading is a necessary step prior to executing as shown in the intrinsic and extrinsic evidence.

At a minimum, this Court should clarify that execution does not mean loading of the software application, but instead takes place while running the instructions within the application after it has been loaded. If the term is not construed now, Blue Spike will mislead the jury at trial saying that the claim term “during execution of the software application” merely means “loading of the software application.” This will only lead to jury confusion. *O2 Micro Int’l Ltd.*, 521 F.3d at 1360 (“When the parties raise an actual dispute regarding the proper scope of these claims, the court, not the jury, must resolve that dispute.”).

Toshiba’s construction should be adopted because the words of the asserted claim, the specification of the Asserted Patents, and extrinsic sources available as of the filing date of the patent – all confirm that loading and execution are separate operations.

e. “memory scheduler”; “memory scheduler code resource”

Defendants’ Construction	Blue Spike’s Construction
<u>Proposed construction:</u> Means-plus-function limitation Claimed Function: During execution time (i) shuffle or randomize other code resources in memory; (ii) modify a stack frame in memory; (iii) modify a value stored by a program counter; (iv) modify a calling address; (v) copy itself to a memory location associated with a calling address; (vi) maintain a list of addresses; (vii) shuffle itself randomly in memory. There must be a disclosed algorithm for performing the recited functions; if not, then it is indefinite. No corresponding structure is disclosed; there is no algorithm; therefore, the term is	<u>Proposed construction:</u> Plain and ordinary

indefinite.	
Alternate construction: memory scheduler, wherein the memory scheduler is within the/a software application	

Toshiba contends that the “memory scheduler code resource” and “memory scheduler” limitations in the ‘569 and ‘719 patents are means-plus-function limitations for which both patents fail to disclose any corresponding structure.

The standard for determining whether a limitation is a means-plus-function limitation “is whether the words of the claim are understood by persons of ordinary skill in the art to have a sufficiently definite meaning as the name for structure.” *Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1349 (Fed. Cir. 2015). The absence of the word “means” does not exclude a term from the requirements of § 112 paragraph 6. “When a claim term lacks the word ‘means,’ the presumption [that § 112 paragraph 6 does not apply] can be overcome . . . if the challenger demonstrates that the claim term fails to ‘recite sufficiently definite structure’ or else recites ‘function without reciting sufficient structure for performing that function.’” *Id.*

In *Williamson*, the disputed term was “distributed learning control module” and the Court found that “module” was a nonce word that failed to identify any “sufficiently definite structure” and instead functioned as a black box just as if the term “means” had been used. *Id.* at 17. This analysis applies equally to “memory scheduler.” The patents’ only description of the “memory scheduler” black box occurs at 8:1-19 of the ‘569 Patent where the term is first introduced as a “*special*” code resource. The inventors provide a name for this resource through the use of quotation marks: “memory scheduler.” This usage shows that “memory scheduler” is a new term with no established meaning or structure and effectively functions here as a nonce word. A term that is set off by quotation marks is often a strong indication that what follows is a

definition. *See, e.g., Cultor Corp. v. A.E. Staley Mfg. Co.*, 224 F.3d 1328, 1331 (Fed. Cir. 2000) (finding that the claim term “water-soluble polydextrose” was expressly defined in the specification). In fact, the claim term could equally have been written as “memory scheduler means.”

The term is also in a format that is “consistent with traditional means-plus-function claim limitations. *Id.* at 17. In such a format, the nonce term recites functions that it performs. That format appears in many places in the Asserted Patents, such as claim 11 of the ‘719 patent: “wherein said memory scheduler [means] is designed to shuffle said other executable code resources. . . .” ‘719 Patent at 17:3-5.

Blue Spike’s proffered definition of “scheduler” also supports the nonce conclusion. That definition is: “a computer program designed to perform functions such as” D.I. 57 at 16; D.I. 76 at 13. In the context of patents on software technology, this definition confirms that “memory scheduler” has no “well understood structural meaning in the computer technology field.” *Williamson*, 792 F.3d at 1359-60. In other words, it is a black box that connotes no structure.

Blue Spike’s brief utterly fails to address this issue, stating only that “memory scheduler” would be “readily understood” by a person of skill. Brief at 13.⁹ In *Williamson*, the patentee offered an expert declaration to show how the disputed term described structure. The Federal Circuit rejected that explanation because it failed to explain **how** the term would be understood to be a “name for structure.” *Williamson*, 792 F.3d at 1362-63.

Here, there is no expert explanation of **how** “memory scheduler” describes structure, only a pathetic reference to a dictionary definition of a scheduler as a computer program that performs

⁹ Blue Spike has no excuse for failing to address this issue. It was fully informed by the parties’ joint claim construction chart that Toshiba sought a means-plus-function construction. *See* D.I. 57.

functions. But, the performance of functions by a computer program is nothing more than a generic phrase; Blue Spike's proffered definition is meaningless and underscores the emptiness of this term.

Once a limitation is found to be subject to application of § 112 paragraph 6, the Court determines "whether the specification discloses sufficient structure that corresponds to the claimed function." *Id.* At the claim construction phase, the Court must "first identify the claimed function" and then "determine what structure, if any, disclosed in the specification corresponds to the claimed function." *Noah Sys., Inc. v. Intuit Inc.*, 675 F.3d 1302, 1311 (Fed. Cir. 2012).

For the first step, the claimed functions of the "memory scheduler code resource" and "memory scheduler" limitations are during execution time to: (i) intermittently relocate, shuffle or randomize other code resources in memory ('569 Patent, claim 18; '719 Patent, claims 10, 11, 12, 18, 23); (ii) modify a stack frame in memory ('719 Patent, claim 23); (iii) modify a value stored by a program counter ('719 Patent, claim 13); (iv) modify a calling address ('569 Patent, claim 17; '719 Patent, claim 5¹⁰); (v) copy itself to a memory location associated with a calling address ('719 Patent, claim 5¹⁰); (vi) maintain a list of addresses ('719 Patent, claims 14, 15); (vii) shuffle itself randomly in memory ('569 Patent, claim 18; '719 Patent, claim 19¹⁰); and (viii) receive calls ('569 Patent, claim 18; '719 Patent, claims 16, 17); *See also* '569 Patent, 8:1-19.

Neither specification provides corresponding structure for these various claimed functions. The entire discussion of "memory scheduler" appears in the '569 Patent at 8:1-19.¹¹

¹⁰ Claim 5 is not asserted, but is provided here for context of the claimed functions.

¹¹ The patentees copied this section into the '719 patent and it is that patent's sole disclosure regarding "memory scheduler." *See* '719 Patent at 15:36-53.

This disclosure identifies numerous functions, such as intentionally shuffling other code resources in memory, but it reveals no structure.¹² Both patents make plain that the invention is to be used in “general computing devices,” further reinforcing the need for corresponding structure. ‘569 Patent at 7:59-62. Because the memory scheduler performs specialized functions that “must be implemented in a special purpose computer,” (*Williamson*, 792 F.3d at 21) the specification must disclose an algorithm for performing the claimed function. *Williamson*, 792 F.3d at 22. Neither patent discloses such an algorithm. The term “memory scheduler” is therefore indefinite because it lacks any corresponding structure. *Triton Tech of Texas, LLC v. Nintendo of Am., Inc.*, 753 F.3d 1375, 1380 (Fed. Cir. 2014) (“Triton was required to disclose an algorithm for performing the claimed integrating function. Because it did not do so, the asserted claims are indefinite”). *MyMedicalRecords, Inc. v. Walgreen Co.*, No. 2:13-CV-00631-ODW, 2014 WL 4367949, at *6 (C.D. Cal. Sept. 3, 2014).

Alternatively, if this Court does find sufficient structure is recited and proceeds to construe the limitation, Toshiba proposes that this Court construe the limitations to mean “memory scheduler, wherein the memory scheduler is within the/an application” as recited, for example, in the specification’s only paragraph describing the term: “**the application contains a special code resource** which knows about all the other code resources in memory . . . this special code resource, called a ‘**memory scheduler**’” ‘569 Patent at 8:1-4.

f. “call”

Defendants’ Construction	Blue Spike’s Construction
<u>Proposed Construction:</u>	<u>Proposed Construction:</u>

¹² Specification disclosed functions: i) shuffle or randomize other code resources in memory; (ii) modify a stack frame in memory; (iii) modify a value stored by a program counter; (iv) modify a calling address; (v) copy itself to a memory location associated with a calling address; (vi) maintain a list of addresses; (vii) shuffle itself randomly in memory.

“Transfer of control during execution time.”	Plain and ordinary <u>Alternate construction:</u> “A statement in a computer program that references a subroutine or program.”
--	--

Toshiba proposes that “call” be construed to reflect the meaning set forth in the specification of the ‘569 Patent. Each of the Asserted claims use “call” in reference to the “memory scheduler.” (See claims 17-18 of the ‘569 Patent and claims 16-17 of the ‘719 Patent). The specification explicitly states that “memory scheduler” is called during execution time: “**during execution time**, this special code resource, called a ‘memory scheduler’ can be **called** . . .”. ‘569 Patent at 8:3-4 (emphasis added). The patent specification thus provides the context as to when the call occurs – during execution time.

Further, the extrinsic evidence supports that “call” should be construed to mean a transfer of control. For example, the McGraw Hill Dictionary of Scientific and Technical Terms, Sixth Ed., 2003, at 315 defines “call” as “to transfer control to a specified closed subroutine.” Ex. 14 Additionally, Oxford Dictionary of Computing, 6th Ed., 2008, at 63 defines “call” as “to transfer control to a subroutine or procedure. . . .” Ex. 15

Blue Spike dictionary evidence, which is cited in the joint claim construction statement, is consistent. (See Ex. 16 IEEE Standard Dictionary of Electrical and Electronics Terms, 6th Ed., 1997 (call: “A transfer of control from one software module to another, usually with the implication that control will be returned to the calling module, *Contrast*: go to.”). This definition refutes Blue Spike’s assertion that “call” requires no construction because a person of skill in the art would understand it to mean “that when a program is called it is triggered to perform its assigned function” is flawed.

g. “intermittently relocating”; “intermittently”; “relocating”

Defendants' Construction	Blue Spike's Construction
<p><u>Proposed construction:</u></p> <p>“intentionally shuffling at periodic, random or pseudo-random intervals”</p>	<p><u>Proposed construction:</u> Plain and ordinary</p> <p><u>Alternate construction 1:</u> “relocating one or more times”</p> <p><u>Alternate construction 2:</u> “moving to a new location one or more times”</p>

Toshiba’s proposed construction is this Court’s previous construction in *Huawei* but a dispute still exists as to whether intermittently means “shuffling” that occurs more than one time.

Judge Schroeder wrote that the prosecution history is ambiguous as to whether the term “intermittently relocating” requires more than one shuffle. Ex. 4 at 9. He then noted that the patentee addressed that ambiguity by stating that shuffling occurs at “intervals” and this Court evidently concluded from that statement that intermittently relocating at intervals could include only one shuffle.

Toshiba respectfully submits that this construction and the claim language require more than one shuffle. First, the claim phrase is “intermittently relocating.” The word “relocating” means locating from an original location; i.e. locating again. This describes a single shuffle. Had the patentee wished to describe a single shuffle, no further wording would have been necessary and under well-established claim construction principles, multiple shuffles would have been encompassed by claiming a single shuffle. A construction that includes one shuffles reads “intermittently” out of the claims. *Enzo Biochem Inc. v. Applera Corp.*, 780 F.3d 1149, 1154 (Fed. Cir. 2015); *See Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed.Cir.2006) (“Claims are interpreted with an eye toward giving effect to all terms in the claim.”). Because repeated movement of code resources is central to this invention, the patentee added the word “intermittently” to reflect multiple, spaced shuffles. This is also reflected in the specification’s

discussion about moving such resources at “intervals.” ‘569 Patent at 8:3-7. In the prosecution history, the patentee reiterated this interval language when faced with an indefiniteness.¹³

“Intervals,” as used in the specification, in the Applicants’ statement to the examiner, and in Judge Schroeder’s opinion, is in the plural form, thus conveying the meaning of more than one interval. And an interval is defined by two points, in this case at least a first location and then a first shuffle. Intervals in the plural form unambiguously conveys a first location and two shuffles.

Blue Spike makes two complaints about Toshiba’s proposed construction. First, it appears to complain that Toshiba did not include “periodic” as another form of interval. Toshiba does not object to this addition to its construction, especially since it only repeats the patentee’s own words. Second, Blue Spike complains that “intentionally” refers to the activity of the memory scheduler, which is not part of claim 16. Blue Spike’s argument ignores the fact that the applicants distinguished the present invention from prior art systems that would rearrange memory, especially in low memory system, to ensure optimal memory utilization. ‘569 Patent at 7:30-36. According to the patents, such prior art reorganizations by the operating system were not an intentional effort to secure copy protection, unlike the present invention. A construction that did not include the intentional aspect of this invention would not capture the Applicants’ own intent to describe an intentional intermittent relocating. Furthermore, Blue Spike does not appear to dispute that “intentionally” is properly applied to claims with the memory scheduler element, which includes four claims that Blue Spike has asserted which depend from Claim 16. It would be difficult to explain to the jury why “intentionally” is a proper construction for the term when applied to the dependent claims, but is not proper when applied to the independent

¹³ Ex. 17 ‘569 Patent prosecution history, Response to Final Office action received June 26, 1997 at 7.

claim, especially when the inventors made no such distinction in the specification. This Court should reject Blue Spike's argument on this part of the construction

Blue Spike's argument that the term is "readily understood" is also incorrect. Even the examiner needed clarification as to what "intermittently relocating" meant and this Court also noted its "inherent ambiguity." (Ex. 18 '569 Patent prosecution history Office Action May 22, 1997 at 2) ("it is unclear what is meant by 'relocating' and by 'intermittently relocating' in context."); *See* Ex. 4 at 9. Further, Blue Spike's proposed alternate construction is presented without any support. Blue Spike's only claim is that its proposed alternate construction "does more work" than Toshiba's proposed construction.

h. "shuffle" "randomize" "relocate"

Defendants' Construction	Blue Spike's Construction
Shuffle <u>Proposed construction:</u> "Randomly reorganize during program run time." Randomize <u>Proposed construction:</u> "Randomly reorganize during program run time." Relocate <u>Proposed construction:</u> "Randomly reorganize during program run time."	Shuffle <u>Proposed construction:</u> Plain and ordinary <u>Alternate construction:</u> "move a portion of a sequence to a new location in a sequence" Randomize <u>Proposed construction:</u> "employ random selection" Relocate <u>Proposed construction:</u> "move to a new location"

Toshiba combines for purposes of simplicity in this brief the arguments for the terms "shuffle," "randomize," and "relocate" because Toshiba's proposed construction (supported by

the intrinsic record) is identical for these three terms. Although as a default, different terms have different meanings (*Applied Med. Res. Corp. v. U.S. Surgical Corp.*, 448 F.3d 1324, 1333 n. 3 (Fed. Cir. 2006)) a Court can construe different terms to have the same meanings where the patent and prosecution history does not reveal a difference between the terms. *Fargo Elecs., Inc. v. Iris Ltd., Inc.*, No. 04-1017 JRT/FLN, 2005 WL 3241851, at *14 (D. Minn. Nov. 30, 2005), *aff'd*, 287 F. App'x 96 (Fed. Cir. 2008) (“In so doing, the Court is aware that it is defining different terms to have essentially the same meaning. Although a patentee's use of different terms normally indicates that it intended those terms to carry different meanings, a reading of the patent and prosecution history does not reveal what that difference might be in this case”). Here, the patent prosecution history and the patent itself do not show a difference between the terms “shuffle,” “randomize,” and “relocate” and thus should be construed with the same meaning.

The specification of the ‘569 patent supports the construction of all three terms to mean to “randomly reorganize during program run time.” The core of the invention disclosed in the ‘569 Patent is the shuffling of code resources randomly in memory to prevent unauthorized copying and increase protection. This shuffling process is referred to in many ways throughout the patent – relocate, randomize, shuffle – but all of these terms have the same meaning. Indeed, the specification of the ‘569 patent recites: “It is also desirable to **randomly reorganize** program memory structure intermittently **during program run time**, to prevent attempts at memory capture or object code analysis aimed at eliminating licensing or ownership information, or otherwise modifying, in an unintended manner, the functioning of the application.” ‘569 Patent at 2:21-26 (emphasis added). The specification specifically identifies what is meant by

shuffling, relocating and randomizing (random reorganization)¹⁴ and when it happens (during program time). ‘569 Patent at 2:21-26. The specification further confirms that this process occurs “during execution time” – thus when the program is running. ‘569 Patent at 8:3-7 (**“During execution time, this special code resource, called a "memory scheduler," can be called periodically, or at random or pseudo random intervals, at which time it intentionally **shuffles the other code resources randomly in memory . . .**”**)(emphases added). Indeed, even the applicants equated the terms during prosecution in order to obtain allowance: “[the claims] stand rejected in further part because ‘it is unclear what is meant by **‘relocating’** and by ‘intermittently relocating’ in context’ . . . Once again, the Applicants respectfully direct the Examiner’s attention to the disclosure at [‘569 patent, 8:1-19]¹⁵, which describes an embodiment of the present invention wherein code resources are **‘shuffled’** in memory at **periodic, random, or pseudo-random intervals.**” (Ex. 17 at 7.)

Nothing in the patent specification, patent claims, or the prosecution history show that these terms are to be treated differently. To the contrary, the intrinsic record confirms that all three terms should be construed to mean to “randomly reorganize during program run time.” Blue Spike’s allegations that (1) Toshiba has improperly suggested that shuffling be at random and that (2) that randomizing does not need to occur during program run time are absurd. The drafters of the patent explicitly refer to randomization in the context of shuffling. ‘569 Patent at 8:3-7. Further, the drafters state in the specification that random reorganization occurs “during program run time.” ‘569 Patent at 2:21-26. For all three of these terms, Blue Spike claims no construction is needed, and then in an afterthought proposes an alternate construction without

¹⁴ Reference to random reorganization appears multiple times in the specification. See, e.g., ‘569 Patent at 7:22-24 (“A second method according to the present invention **is to randomly re-organize** program memory structure to prevent attempts at memory capture or object code analysis.”)(emphasis added).

¹⁵ Toshiba notes that this referenced disclosure too references operation during runtime.

any basis or support for it. Toshiba's proposed construction for "shuffle," "randomize," and "relocate" is supported by the intrinsic record, nothing in the record indicates that these terms were to be construed with different meanings, and as such this Court should adopt Toshiba's construction.

V. CONCLUSION

Toshiba respectfully requests that this Court adopt its proposed constructions, and find the terms "memory scheduler" and "memory scheduler code resource" indefinite for failing to disclose a corresponding structure for the claimed function.

Dated: April 10, 2017

Respectfully submitted,

By: /s/ Melissa R. Smith

Melissa R. Smith
Texas Bar No. 24001351
Melissa@gillamsmithlaw.com
GILLAM & SMITH, LLP
303 South Washington Avenue
Marshall, Texas 75670
Telephone: (903) 934-8450
Facsimile: (903) 934-9257

Jeffrey K. Sherwood
Texas Bar No. 24009354
JSherwood@BlankRome.com
Charles J. Monterio, Jr.
DC Bar No. 495483
CMonterio@BlankRome.com
Alexander S. Perry
DC Bar No. 1010393
APerry@BlankRome.com
Gabriella E. Zicarelli
DC Bar No. 1023338
GZicarelli@BlankRome.com
BLANK ROME LLP
1825 Eye Street, NW
Washington, DC 20006
Telephone: (202) 420-2200
Facsimile: (202) 420-2201

Gerard A. Haddad
New York Bar No. 2731271
GHaddad@BlankRome.com
BLANK ROME LLP
405 Lexington Avenue
New York, NY 10174-0208
Telephone: (212) 885-5135
Facsimile: (917) 591-6921

Attorneys for Defendants
Toshiba America Information Systems, Inc.,
and Toshiba Corporation

CERTIFICATE OF SERVICE

I hereby certify that on April 10, 2017, a true and correct copy of the foregoing document was filed electronically in compliance with Local Rule CV-5. As of this date, all counsel of record have consented to electronic service and are being served with a copy of this document through the Court's CM/ECF system under Local Rule CV-5(a)(3)(A).

/s/ Melissa R. Smith

Melissa R. Smith